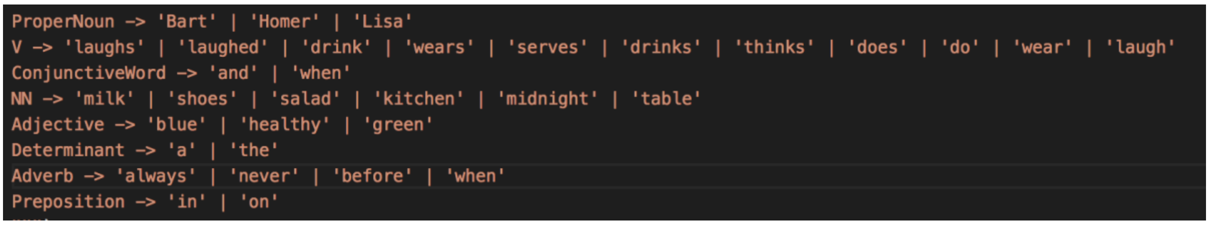
STEP 1: Lexicon

The parts of speech in the example sentences were broken down as such and a lexicon was made. Attached below as Figure 1 is the Lexicon.

The POS tags that were used were mostly unconventional and hence, an APPENDICES which serves as a POS Tag key, has been attached to reference the tag naming convention used in this practical.

- Proper Noun (‘Bart’), Verb (‘laughs’)
- Proper Noun (‘Homer’), Verb (‘laughed’)
- Proper Noun (‘Bart’), Conjunction (‘and’), Proper Noun (‘Lisa’), Verb (‘drink’), Noun (‘’Milk)
- Proper Noun (‘Bart’), Verb (‘wears’), Adjective (‘blue’), Noun (‘shoes’)
- Proper Noun (‘Lisa’), Verb (‘serves’), Proper Noun (‘Bart’), Determinant (‘a’), Adjective (‘healthy’), Adjective (‘green’), Noun (‘salad’)
- Proper Noun (‘Homer’), Verb (‘serves’), Proper Noun (‘Lisa’)
- Proper Noun (‘Bart’), Adverb (‘always’), Verb (‘drinks’), Noun (‘milk’)
- Proper Noun (‘Lisa’), Verb (‘thinks’), Proper Noun (‘Homer’), Verb (‘thinks’), Proper Noun (‘Bart’), Verb (‘drinks’), Noun (‘milk’)
- Proper Noun (‘Homer’), Adverb (‘never’), Verb (‘drinks’), Noun (‘milk’), Preposition (‘in’), Determinant (‘the’), Noun (‘kitchen’), Adverb (‘before’), Noun (‘midnight’)
- Conjunction (‘when’), Proper Noun (‘Homer’), Verb (‘drinks’), Noun (‘milk’), Proper Noun (‘Bart’), Verb (‘laughs’)
- Adverb (‘when’), Verb (‘does’), Proper Noun (‘Lisa’), Verb (‘drink’), Determinant (‘the’), Noun (‘milk’), Preposition (‘on’), Determinant (‘the’), Noun (‘table’)
- Adverb (‘when’), Verb(‘do’), Proper Noun (‘Lisa’), Conjunction (‘and’), Proper Noun (‘Bart’), Verb (‘wear’), Noun (‘shoes’)

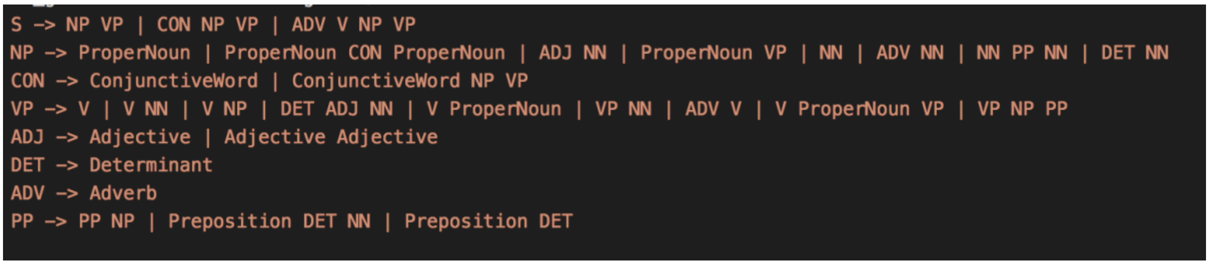


```
ProperNoun -> 'Bart' | 'Homer' | 'Lisa'
V -> 'laughs' | 'laughed' | 'drink' | 'wears' | 'serves' | 'drinks' | 'thinks' | 'does' | 'do' | 'wear' | 'laugh'
ConjunctiveWord -> 'and' | 'when'
NN -> 'milk' | 'shoes' | 'salad' | 'kitchen' | 'midnight' | 'table'
Adjective -> 'blue' | 'healthy' | 'green'
Determinant -> 'a' | 'the'
Adverb -> 'always' | 'never' | 'before' | 'when'
Preposition -> 'in' | 'on'
```

Figure 1 - Lexicon

STEP 2: Context-free Grammar

Context free rules were then created and added. Figure 2 shows the final context-free grammar rules that were used to parse the example sentences.



```
S -> NP VP | CON NP VP | ADV V NP VP
NP -> ProperNoun | ProperNoun CON ProperNoun | ADJ NN | ProperNoun VP | NN | ADV NN | NN PP NN | DET NN
CON -> ConjunctiveWord | ConjunctiveWord NP VP
VP -> V | V NN | V NP | DET ADJ NN | V ProperNoun | VP NN | ADV V | V ProperNoun VP | VP NP PP
ADJ -> Adjective | Adjective Adjective
DET -> Determinant
ADV -> Adverb
PP -> PP NP | Preposition DET NN | Preposition DET
```

Figure 2 – Context-Free Grammar Rules

STEP 3: Intermediate Testing of the Grammar
The intermediate test was then undertaken with the 'cf_parser' in the main.py to run and the acceptance of the grammar using the example 12 sentences.

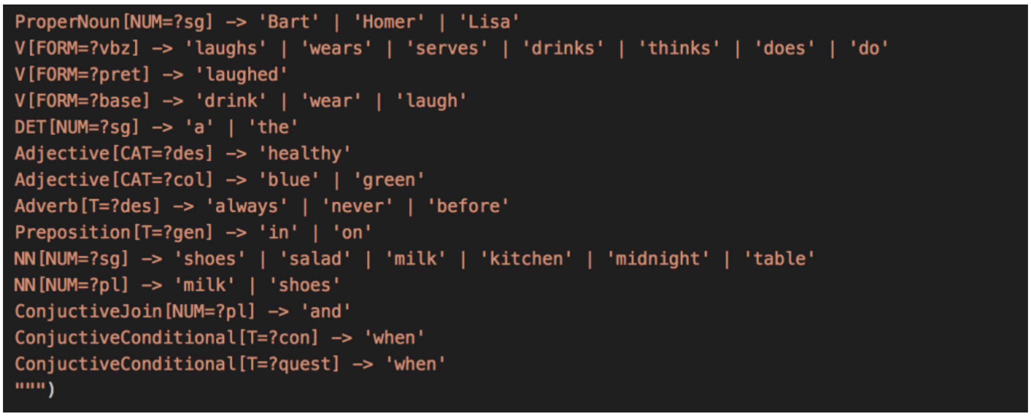
The sentences were parsed successfully. With a few observations made, detailed in the following points:

- Some sentences with more than two words i.e. more than just a noun and a terminal verb, had multiple trees generated.
- This indicated some overlap and also the idea of grammar in the English language having endless possibilities of construction.
- It also suggested the idea that certain words in the language have multiple meanings and use.

STEP 4: Unification Grammar

This part involved modifying the CFG[1] parser to take in arguments in order enable some contextual processing, i.e. differentiating singular nouns from plural OR differentiating between present and past verb tenses.

The lexicon of the CFG was modified to include these distinctions. For example, 'laughs' is of the from 'vbz' while 'laughed' is of the 'pret' form. The lexicon is attached below as Figure xx.



```
ProperNoun[NUM=?sg] -> 'Bart' | 'Homer' | 'Lisa'
V[FORM=?vbz] -> 'laughs' | 'wears' | 'serves' | 'drinks' | 'thinks' | 'does' | 'do'
V[FORM=?pret] -> 'laughed'
V[FORM=?base] -> 'drink' | 'wear' | 'laugh'
DET[NUM=?sg] -> 'a' | 'the'
Adjective[CAT=?des] -> 'healthy'
Adjective[CAT=?col] -> 'blue' | 'green'
Adverb[T=?des] -> 'always' | 'never' | 'before'
Preposition[T=?gen] -> 'in' | 'on'
NN[NUM=?sg] -> 'shoes' | 'salad' | 'milk' | 'kitchen' | 'midnight' | 'table'
NN[NUM=?pl] -> 'milk' | 'shoes'
ConjuctiveJoin[NUM=?pl] -> 'and'
ConjuctiveConditional[T=?con] -> 'when'
ConjuctiveConditional[T=?quest] -> 'when'
""")
```

Figure 3 – Unification Feature Grammar

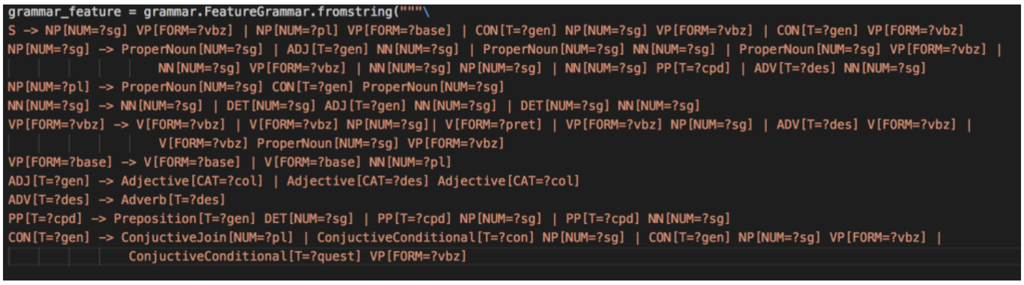
The implementation was made in a way that each POS tag takes an argument for a subcategory, which can be then used to identify each word for the appropriate context within the grammar of the sentence. For example, identifying the type of verb 'laughs' (i.e. e.g. past or present) is means, identifying it with an argument within the TAG set that can distinguish between the several types of verbs are.

Numeration i.e. number context, a subcategorization of noun tags into singular or plural context was also implemented using the NUM subcategory argument.

[1] Context-Free Grammar

The implementation of the subcategorization in both nouns (i.e. singular or plural) and verbs (i.e. present plural, present singular or past tenses) enable some contextual subject-verb agreement. In essence, the model could then map singular nouns to singular verbs (and can differentiate between past or present).

The rules of the unification model were updated with the subcategories within the POS tags as illustrated in Figure xx.



```
grammar_feature = grammar.FeatureGrammar.fromstring("""\
S -> NP[NUM=?sg] VP[FORM=?vbz] | NP[NUM=?pl] VP[FORM=?base] | CON[T=?gen] NP[NUM=?sg] VP[FORM=?vbz] | CON[T=?gen] VP[FORM=?vbz]
NP[NUM=?sg] -> ProperNoun[NUM=?sg] | ADJ[T=?gen] NN[NUM=?sg] | ProperNoun[NUM=?sg] NN[NUM=?sg] | ProperNoun[NUM=?sg] VP[FORM=?vbz] |
                    NN[NUM=?sg] VP[FORM=?vbz] | NN[NUM=?sg] NP[NUM=?sg] | NN[NUM=?sg] PP[T=?cpd] | ADV[T=?des] NN[NUM=?sg]
NP[NUM=?pl] -> ProperNoun[NUM=?sg] CON[T=?gen] ProperNoun[NUM=?sg]
NN[NUM=?sg] -> NN[NUM=?sg] | DET[NUM=?sg] ADJ[T=?gen] NN[NUM=?sg] | DET[NUM=?sg] NN[NUM=?sg]
VP[FORM=?vbz] -> V[FORM=?vbz] | V[FORM=?vbz] NP[NUM=?sg]| V[FORM=?pret] | VP[FORM=?vbz] NP[NUM=?sg] | ADV[T=?des] V[FORM=?vbz] |
                    V[FORM=?vbz] ProperNoun[NUM=?sg] VP[FORM=?vbz]
VP[FORM=?base] -> V[FORM=?base] | V[FORM=?base] NN[NUM=?pl]
ADJ[T=?gen] -> Adjective[CAT=?col] | Adjective[CAT=?des] Adjective[CAT=?col]
ADV[T=?des] -> Adverb[T=?des]
PP[T=?cpd] -> Preposition[T=?gen] DET[NUM=?sg] | PP[T=?cpd] NP[NUM=?sg] | PP[T=?cpd] NN[NUM=?sg]
CON[T=?gen] -> ConjuctiveJoin[NUM=?pl] | ConjuctiveConditional[T=?con] NP[NUM=?sg] | CON[T=?gen] NP[NUM=?sg] VP[FORM=?vbz] |
                ConjuctiveConditional[T=?quest] VP[FORM=?vbz]
```

Figure 4 - Unification Model

This unification model, i.e. 'grammar_feature', was used to parse the twelve example sentences and parsed them successfully. Similar observations to the previous CFG model was made where, multiple trees were created for sentences.

STEP 5: Testing and Evaluation

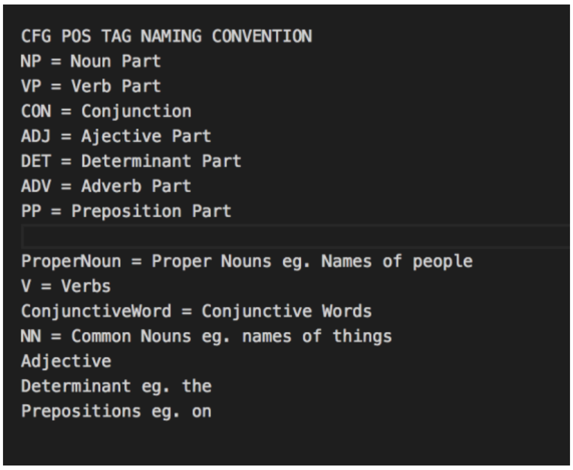
The two models were both tested on the three grammatically wrong, given sentences in the practical specification and these observations were made:

- Both models accepted all sentences tested and also mapped out their trees.
- The unification model, was intended to reject the 'wrong sentences' and however it accepted them.
- Both models predicted multiple trees for each sentence
- The unification model did show the right trees and tagged the words within the sentences with the right subcategories.
- The unification model struggled to reject grammatically wrong sentences.

Overall the practical was interesting as it taught several ways of implementing grammar, be it contextual or non-contextual, into digital systems. It was however learnt that, a good grammar engineering need really good, simply models that can interpret context, in order to distinguish between words within a lexicon based on their POS.

APPENDICES

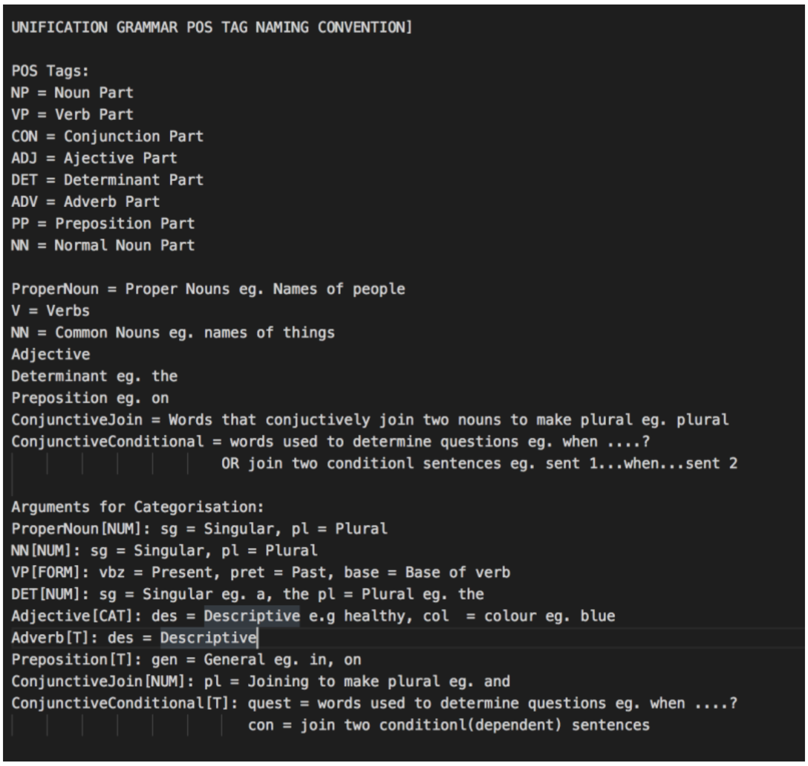
Appendix 1



```
CFG POS TAG NAMING CONVENTION
NP = Noun Part
VP = Verb Part
CON = Conjunction
ADJ = Ajective Part
DET = Determinant Part
ADV = Adverb Part
PP = Preposition Part

ProperNoun = Proper Nouns eg. Names of people
V = Verbs
ConjunctiveWord = Conjunctive Words
NN = Common Nouns eg. names of things
Adjective
Determinant eg. the
Prepositions eg. on
```

Appendix 2



```
UNIFICATION GRAMMAR POS TAG NAMING CONVENTION]

POS Tags:
NP = Noun Part
VP = Verb Part
CON = Conjunction Part
ADJ = Ajective Part
DET = Determinant Part
ADV = Adverb Part
PP = Preposition Part
NN = Normal Noun Part

ProperNoun = Proper Nouns eg. Names of people
V = Verbs
NN = Common Nouns eg. names of things
Adjective
Determinant eg. the
Preposition eg. on
ConjunctiveJoin = Words that conjuctively join two nouns to make plural eg. plural
ConjunctiveConditional = words used to determine questions eg. when ....?
                        OR join two conditionl sentences eg. sent 1...when...sent 2

Arguments for Categorisation:
ProperNoun[NUM]: sg = Singular, pl = Plural
NN[NUM]: sg = Singular, pl = Plural
VP[FORM]: vbz = Present, pret = Past, base = Base of verb
DET[NUM]: sg = Singular eg. a, the pl = Plural eg. the
Adjective[CAT]: des = Descriptive e.g healthy, col  = colour eg. blue
Adverb[T]: des = Descriptive
Preposition[T]: gen = General eg. in, on
ConjunctiveJoin[NUM]: pl = Joining to make plural eg. and
ConjunctiveConditional[T]: quest = words used to determine questions eg. when ....?
                           con = join two conditionl(dependent) sentences
```